Algoritma Sederhana dalam Memahami Proses Pengurutan Data

Eko Nur Wahyudi
Fakultas Teknologi Informasi, Universitas Stikubank Semarang
eko@unisbank.ac.id

Abstrak : Pengurutan data atau sorting merupakan suatu proses dimana suatu susunan data yang semula dalam kondisi acak dapat menjadi urut, baik dari data terkecil sampai dengan data yang terbesar, atau sebaliknya dari data terbesar sampai dengan data terkecil. Terdapat beberapa metode yang cukup populer dalam proses pengurutan data tersebut. Berikut ini akan dijelaskan bagaimana proses pengurutan data masing-masing metode tersebut dengan menggunakan algoritma secara sederhana.

Kata kunci : Sorting, Pengurutan data, Algoritma

PENDAHULUAN

1. Metode Selection

Pengurutan data dengan metode Selection dapat dijelaskan secara sederhana yaitu dengan cara mencari data terkecil dari susunan data awal dari data pertama sampai dengan data terakhir. Setelah data terkecil ditemukan kemudian tukarkan dengan data pertama. Setelah itu dicari lagi data terkecil tetapi mulai dari data kedua sampai data terakhir dan setelah didapatkan kemudian tukarkan dengan data kedua. Begitu seterusnya hingga diperoleh data urut dari awal sampai akhir.

Contoh :

Data awal :
30 70 10 40 60 20 50

Cari data terkecil dari data pertama sampai data terakhir. Ditemukan data 10 pada urutan ketiga. Tukarkan data 10 dengan data 30 di urutan pertama sehingga hasilnya menjadi seperti berikut :

10 70 30 40 60 20 50

Selanjutnya cari data terkecil mulai dari urutan data kedua sampai data terakhir. Ditemukan data 20 di urutan keenam. Tukarkan data 20 dengan data 70 di urutan kedua sehingga hasilnya berubah menjadi seperti berikut :

10 20 30 40 60 70 50

Cari kembali data terkecil mulai dari urutan data ketiga. Ditemukan data 30 yang sudah pada urutan ketiga sehingga tidak terjadi pertukaran posisi dengan data yang lain. Hasilnya adalah seperti berikut :

10 20 30 40 60 70 50

Pada pencarian data terkecil keempat hasilnya sama seperti saat pencarian data terkecil ketiga dimana data 40 sudah berada pada urutan data keempat dan tidak mengalami pertukaran data. Hasilnya adalah seperti berikut :

10 20 30 40 60 70 50

Berikutnya pencarian data terkecil dimulai dari urutan data kelima. Ditemukan data 50 di urutan data terakhir yang kemudian ditukar dengan data urutan kelima sehingga hasilnya menjadi seperti berikut :

10 20 30 40 50 70 60

Pencarian data terkecil terakhir dilakukan mulai urutan data keenam, dimana data terakhir yaitu 60 harus ditukar dengan data 70 yang berada pada urutan keenam sehingga hasilnya menjadi seperti berikut :

10 20 30 40 50 60 70

Data dalam keadaan urut secara ascending.

Untuk mendapatkan urutan data yang sebaliknya secara descending, yaitu urutan dari data terbesar ke data terkecil, cukup dengan merubah nilai data yang dicari menjadi data yang terbesar.

2. Metode Straight Insertion

Pengurutan data dengan metode Straight Insertion ini menggunakan cara seperti halnya orang bermain kartu, dimana susunan kartu dibaca dari kiri ke kanan dan selalu dibandingkan dengan susunan kartu yang telah dibaca sebelumnya. Apabila kartu yang sedang dibaca nilainya lebih kecil dari kartu yang dibandingkan kemudian disisipkan di posisi sebelah kiri kartu tersebut dan menggeser kartu yang lebih besar ke sebelah kanan, proses ini dilakukan mulai dari kartu pertama sampai akhir susunan kartu sehingga diperoleh susunan kartu yang urut.

Contoh :

Data awal :
30 70 10 40 60 20 50

Baca data pertama dari susunan kartu, karena tidak ada urutan data sebelumnya maka data pertama tidak dapat dibandingkan sehingga tetap pada posisi awal dan urutan masih tidak berubah menjadi seperti berikut :

30 70 10 40 60 20 50

Baca data kedua, bandingkan data kedua dengan data pertama yang telah dibaca, apabila nilai data kedua lebih kecil maka sisipkan data kedua pada urutan pertama dan geser data pertama ke urutan kedua. Karena 70 lebih besar dari 30 maka susunan data tetap seperti berikut :

30 70 10 40 60 20 50

Baca data ketiga, kemudian bandingkan dengan data sebelumnya mulai urutan data pertama, kedua dan seterusnya. Ternyata nilai data ketiga lebih kecil dari data pertama, maka sisipkan data ketiga pada urutan pertama, geser data pertama ke urutan kedua dan geser data kedua ke urutan ketiga, sehingga urutan berubah menjadi seperti berikut :

10 30 70 40 60 20 50

Berikutnya baca data keempat dan seperti halnya langkah sebelumnya maka diperoleh urutan menjadi seperti berikut :

10 30 40 70 60 20 50

Selanjutnya baca data kelima, maka akan diperoleh hasil menjadi seperti berikut :

10 30 40 60 70 20 50

Baca data keenam, hasilnya berubah lagi menjadi seperti berikut :

10 20 30 40 60 70 50

Terakhir baca data ketujuh, maka seperti langkah sebelumnya hasilnya akan menjadi seperti berikut :

10 20 30 40 50 60 70

Data urut saat pembacaan data terakhir dan disisipkan pada posisi kelima, menggeser angka 60 dan 70 ke sebelah kanan. Untuk penurutan secara descending cukup dengan mengganti data yang dicari dari yang lebih kecil menjadi yang lebih besar.

3. Metode Binary Insertion

Pengurutan data dengan metode Binary Insertion merupakan metode perbaikan dari metode sebelumnya yaitu Straight Insertion. Kalau pada metode Straight Insertion perbandingan selalu dilakukan dari data pertama maka pada metode Binary Insertion ini perbandingan dilakukan hanya pada kelompok data yang telah dibagi menjadi dua bagian dari susunan data yang telah urut sebelumnya.

Contoh :

Data awal :
30 70 10 40 60 20 50

Baca data pertama, hasilnya akan sama seperti susunan data awal yaitu seperti berikut :

30 70 10 40 60 20 50

Selanjutnya baca data kedua, susunan data masih tidak berubah yaitu seperti berikut :

30 70 10 40 60 20 50

Kemudian baca data ketiga. Data pertama dan kedua yang telah urut dibagi menjadi 2 bagian yaitu 30 dan 70. Data 10 dibandingkan dengan data 30 yang lebih memungkinkan dimana data akan disisipkan daripada data 70. Sisipkan data 10 ke posisi pertama dan menggeser data 30 dan 70 ke kanan seperti berikut :

10 30 70 40 60 20 50

Berikutnya baca data keempat. Data yang telah urut sebelumnya dibagi menjadi dua kelompok bagian yaitu (10 30) dan (70). Data 40 lebih tepat dibandingkan dengan (70) sehingga kelompok data (10 30) bisa diabaikan. Sisipkan data 40 ke urutan 3 dan menggeser data 70 ke sebelah kanan di urutan keempat. Hasilnya menjadi seperti berikut :

10 30 40 70 60 20 50

Baca data kelima. Bagi terlebih dahulu menjadi dua bagian data yang telah urut sebelumnya yaitu menjadi (10 30) dan (40 70). Kemungkinan data 60 disisipkan adalah pada kelompok (40 70) sehingga kelompok (10 30) dapat diabaikan. Bagi kelompok (40 70) menjadi dua bagian lagi yaitu (40) dan (70) untuk lebih tepat dalam membandingkan. Sisipkan data 60 pada urutan keempat dan geser data 70 ke kanan pada urutan kelima. Hasilnya akan menjadi seperti berikut :

10 30 40 60 70 20 50

Berikutnya baca data keenam yaitu 20. Setelah data yang urut sebelumnya dibagi menjadi 2 bagian, yaitu kelompok (10 30 40) dan (60 70), nampaknya kemungkinan perbandingan dilakukan pada kelompok pertama dan kelompok kedua dapat diabaikan. Kelompok data pertama dibagi menjadi dua bagian lagi menjadi (10 30) dan (40). Data (10 30) dibagi lagi menjadi dua bagian (10) dan (30) karena kemungkinan data 20 dibandingkan dan disisipkan ada pada kelompok tersebut. Sisipkan data 20 pada urutan kedua dan geser data 30 ke sebelah kanannya diikuti data 40, 60 dan 70. Hasil pada langkah ini menjadi seperti berikut :

10 20 30 40 60 70 50

Data yang telah urut sebelumnya dibagi menjadi dua bagian, yaitu kelompok (10 20 30) dan (40 60 70). Abaikan kelompok pertama karena kemungkinan data dibandingkan dan disisipkan ada pada kelompok kedua. Bagi lagi kelompok kedua menjadi dua bagian yaitu (40 60) dan (70). Abaikan kelompok data (70) dan bagi lagi kelompok data (40 60) menjadi (40) dan (60). Bandingkan dan sisipkan data 50 pada posisi data 60, geser data 60 dan 70 ke kanan. Data kemudian menjadi urut seperti berikut :

10 20 30 40 50 60 70

Perlu diketahui bahwa antara metode Binary Insertion dan Straight Insertion perbedaannya hanya pada jumlah perbandingannya, sedangkan jumlah pergeserannya akan tetap sama. Pada penyisipan biner perbandingan tidak harus dilakukan dari urutan data pertama, tetapi pada kelompok data dimana kemungkinan data tersebut dibandingkan dan disisipkan. Seperti halnya untuk urutan ascending, jika urutan diinginkan secara descending maka data yang lebih besar yang dicari, dibandingkan dan disisipkan dari kiri ke kanan.

4. Metode BubbleSort

Metode BubbleSort adalah salah satu metode pengurutan data yang menggunakan cara penukaran, atau juga lebih sering disebut dengan metode ExchangeSort. Metode ini mendasarkan pada perbandingan data secara berurut dari awal hingga akhir dan menempatkan data yang lebih besar di sebelah kanan. Jika dalam satu tahap perbandingan belum diperoleh data yang urut maka diulang kembali seperti langkah pertama mulai dari awal tetapi tanpa mengikut sertakan data terakhir karena nilainya sudah yang paling besar. Proses tersebut akan selalu diulang hingga diperoleh urutan data dari kiri ke kanan dari yang paling kecil ke yang paling besar.

Contoh :

Data awal :

30 70 10 40 60 20 50

Bandingkan data pertama dengan data kedua, apabila data pertama lebih besar dari data kedua maka saling bertukar tempat. Pada perbandingan pertama nilai data 30 lebih kecil dari data 70 maka urutan data tidak berubah seperti berikut :

30 70 10 40 60 20 50

Berikutnya bandingkan data kedua dengan data ketiga. Seperti langkah pertama, jika data kedua lebih besar dari data ketiga maka tukarkan tempat. Ternyata nilai data 70 lebih besar dari nilai data 10 sehingga tempatnya harus ditukar, sehingga hasilnya berubah menjadi seperti berikut :

30 10 70 40 60 20 50

Bandingkan data ketiga dan keempat, terjadi perubahan urutan karena nilai data 70 lebih besar dari data 40 sehingga susunan data menjadi seperti berikut :

30 10 40 70 60 20 50

Bandingkan data keempat dan kelima, terjadi perubahan urutan karena nilai data 70 lebih besar dari data 60 sehingga susunan data menjadi seperti berikut :

30 10 40 60 70 20 50

Selanjutnya bandingkan data kelima dan keenam, susunan data berubah menjadi seperti berikut :

30 10 40 60 20 70 50

Bandingkan data keenam dan ketujuh, susunan data berubah lagi menjadi seperti berikut :

30 10 40 60 20 50 70

Setelah membandingkan semua data dari awal hingga akhir secara bertahap, ternyata susunan data masih belum urut seperti yang diharapkan. Untuk itu perlu diulang lagi langkah-langkah diatas, tetapi tanpa membandingkan dengan data terakhir.

Bandingkan data pertama dan kedua, hasilnya menjadi seperti berikut :

10 30 40 60 20 50 70

Bandingkan data kedua dan ketiga, hasilnya adalah tetap seperti berikut :

10 30 40 60 20 50 70

Bandingkan data ketiga dan keempat, hasilnya adalah tetap juga seperti berikut :

10 30 40 60 20 50 70

Bandingkan data keempat dan kelima, hasilnya berubah menjadi seperti berikut :

10 30 40 20 60 50 70

Bandingkan data kelima dan keenam, hasilnya berubah lagi menjadi seperti berikut :

10 30 40 20 50 60 70

Sampai dengan tahap ini data juga belum urut seperti yang diharapkan, langkah di atas diulang kembali tetapi tanpa melibatkan data keenam dan ketujuh yang telah urut dari yang terbesar di sebelah kanan.

Bandingkan data pertama dan kedua, hasilnya adalah seperti berikut :

10 30 40 20 50 60 70

Bandingkan data kedua dan ketiga, hasilnya adalah seperti berikut :

10 30 40 20 50 60 70

Bandingkan data ketiga dan keempat, hasilnya berubah menjadi seperti berikut :

10 30 20 40 50 60 70

Bandingkan data keempat dan kelima, hasilnya tetap seperti berikut :

10 30 20 40 50 60 70

Ulangi lagi langkah awal dengan membandingkan data pertama dan kedua, hasilnya adalah seperti berikut :

10 30 20 40 50 60 70

Bandingkan data kedua dan ketiga, hasilnya berubah menjadi seperti berikut :

10 20 30 40 50 60 70

Sampai dengan langkah ini data sudah dalam keadaan urut, namun jika belum maka proses perbandingan dan penukaran tempat dapat dilakukan sampai langkah terakhir dengan tinggal membandingkan data pertama dan kedua sehingga diperoleh data dalam urutan penuh dari yang paling kecil di kiri dan yang paling besar di kanan. Untuk urutan descending cukup dengan menentukan bahwa apabila data di kiri lebih kecil maka terjadi pertukaran tempat. Hasil akhir yang diperoleh adalah urutan data dari yang terbesar sampai dengan yang terkecil mulai dari kiri ke kanan.

5. Metode Straight Selection

Metode Straight Selection merupakan metode pengurutan data yang hampir sama dengan metode BubbleSort, tetapi pasangan data yang dibandingkan berbeda urutan pembacannya. Pada metode Straight Selection data pertama dibandingkan dengan semua data berikutnya. Apabila data di sebelah kiri yang dibandingkan nilainya lebih besar maka terjadi pertukaran tempat. Berikutnya bandingkan data kedua dengan data berikutnya sampai dengan data terakhir. Proses perbandingan dan penukaran tempat dilakukan seperti proses sebelumnya. Dengan metode ini akan diperoleh data terkecil lebih dahulu di sebelah kiri kemudian bergerak ke kanan ke yang paling besar. Metode Straight Selection merupakan proses kebalikan dari metode BubbleSort yang dimulai dengan mendapatkan nilai data terbesar lebih dahulu di sebelah kanan menuju ke yang terkecil di sebelah kiri.

Contoh :

Data awal :
30 70 10 40 60 20 50

Bandingkan data pertama dan kedua, hasilnya adalah seperti berikut :

30 70 10 40 60 20 50

Bandingkan data pertama dengan data ketiga, terjadi pertukaran tempat dan perubahan urutan menjadi seperti berikut :

10 70 30 40 60 20 50

Bandingkan data pertama dengan data keempat, urutan tetap seperti berikut :

10 70 30 40 60 20 50

Bandingkan data pertama dengan data kelima, hasilnya seperti berikut :

10 70 30 40 60 20 50

Bandingkan data pertama dengan data keenam, hasilnya masih sama seperti berikut :

10 70 30 40 60 20 50

Bandingkan data pertama dengan data terakhir hasilnya seperti berikut :

10 70 30 40 60 20 50

Setelah dilakukan satu tahapan perbandingan dari data pertama sampai dengan data terakhir dan belum diperoleh susunan data yang urut, proses seperti di awal diulang kembali tetapi dimulai dengan membandingkan data kedua dengan data ketiga, hasilnya sebagai berikut :

10 30 70 40 60 20 50

Bandingkan data kedua dengan data keempat, hasilnya tidak berubah seperti berikut :

10 30 70 40 60 20 50

Bandingkan data kedua dengan data kelima, hasilnya tetap seperti berikut :

10 30 70 40 60 20 50

Bandingkan data kedua dengan data keenam, terjadi perubahan susunan seperti berikut :

10 20 70 40 60 30 50

Bandingkan data kedua dengan data ketujuh, hasilnya masih sama seperti berikut :

10 20 70 40 60 30 50

Karena data belum urut maka selanjutnya proses perbandingan data dimulai dari urutan data ketiga dengan data keempat, hasilnya seperti berikut :

10 20 40 70 60 30 50

Bandingkan data ketiga dengan data kelima, hasilnya menjadi seperti berikut :

10 20 40 70 60 30 50

Bandingkan data ketiga dengan data keenam, hasilnya berubah seperti berikut :

10 20 30 70 60 40 50

Bandingkan data ketiga dengan data terakhir dan susunan tidak berubah seperti berikut :

Ulangi proses perbandingan data mulai urutan data keempat karena data belum urut. Bandingkan data keempat dengan data kelima, hasilnya seperti berikut :

10 20 30 60 70 40 50

Bandingkan data keempat dengan data keenam, hasilnya menjadi seperti berikut :

10 20 30 40 70 60 50

Bandingkan data keempat dengan data ketujuh, hasilnya sama seperti berikut :

10 20 30 40 70 60 50

Selanjutnya bandingkan data kelima dengan data keenam, hasilnya adalah seperti berikut :

10 20 30 40 60 70 50

Bandingkan data kelima dengan data ketujuh, hasilnya berubah seperti berikut :

10 20 30 40 50 70 60

Terakhir bandingkan data keenam dengan data ketujuh, hasilnya adalah data yang sudah urut seperti berikut :

10 20 30 40 50 60 70

Untuk urutan secara descending cukup dengan mengganti pilihan bahwa data yang lebih di sebelah kanan akan ditukar tempat dengan data yang lebih kecil di sebelah kiri saat dibandingkan.

6. Metode ShellSort

Metode pengurutan data ini ditemukan oleh tokoh bernama Donald Shell yang kemudian populer dengan istilah Shellsort. Metode ini menggunakan cara dimana suatu susunan data dibagi menjadi beberapa bagian dengan jarak tertentu dan kemudian disusun menjadi bentuk baris dan kolom. Dari tiap kolom kemudian data diurutkan. Apabila jumlah data ganjil maka yang terakhir merupakan sisa pembagian jarak tersebut.

Contoh :

Data awal :
30 60 10 80 90 20 50 40 70

Susunan data awal dengan jarak tertentu, dalam hal ini masing-masing kelompok terdiri dari 4 buah data, kemudian disusun menjadi bentuk baris dan kolom sehingga susunan menjadi seperti berikut :

30 60 10 80 90 20 50 40 70

Berdasarkan susunan kolom tersebut kemudian data diurutkan sehingga hasilnya menjadi seperti berikut :

30 20 10 40 70 60 50 80 90

Data kemudian disusun kembali dalam bentuk baris sehingga urutan menjadi seperti berikut :

30 20 10 40 70 60 50 80 90

Kemudian dengan jarak tertentu lagi yaitu dengan jumlah data separuh dari jarak pada langkah sebelumnya, data disusun dalam bentuk baris dan kolom seperti berikut ini :

30 20
10 40
70 60
50 80
90

Setelah itu sesuai dengan susunan kolomnya data kemudian diurutkan sehingga susunan menjadi seperti berikut :

10 20
30 40
50 60
70 80
90

Setelah susunan data dikembalikan dalam bentuk baris sehingga diperoleh urutan data seperti berikut ini :

10 20 30 40 50 60 70 80 90

Sampai dengan langkah ini data sudah dalam keadaan urut. Apabila diinginkan urutan secara descending maka pada saat proses pengurutan tiap kolom dilakukan dengan nilai dari yang terbesar ke nilai yang terkecil.

7. Metode RadixSort

Dalam metode RadixSort, data yang tersusun harus dikelompokkan berdasarkan urutan digitnya, mulai dari kelompok satuan, puluhan, ratusan dan seterusnya dan disusun dari urutan paling atas ke paling bawah. Selanjutnya berdasarkan digit satuannya kemudian data disusun ulang dengan membaca data dari paling atas ke paling bawah. Hasil dari susunan berdasarkan digit satuan tersebut kemudian disusun kembali secara mendatar. Untuk langkah berikutnya data disusun ke bawah untuk dicari kelompok data berdasarkan digit puluhan nya. Langkah tersebut diulang kembali sampai diperoleh susunan data yang urut.

Contoh :

Data awal :
322 713 122 414 632 221 513

Berdasarkan susunan data tersebut di atas maka sesuai dengan kelompok digit terakhir atau nilai satuan akan dihasilkan susunan data seperti berikut ini :

Digit 1 : 221
Digit 2 : 322 122 632
Digit 3 : 713 513
Digit 4 : 414

Berdasarkan pengelompokan tersebut kemudian data disusun kembali sehingga hasilnya menjadi seperti berikut :

221 322 122 632 713 513 414

Kelompokkan kembali susunan data tersebut di atas berdasarkan digit kedua sehingga menghasilkan kelompok seperti berikut :

Digit 1 : 713 513 414
Digit 2 : 221 322 122
Digit 3 : 632

Sehingga hasil dari pengelompokan tersebut akan merubah susunan data menjadi seperti berikut :

713 513 414 221 322 122 632

Terakhir data dikelompokkan berdasarkan digit pertamanya menjadi seperti berikut :

Digit 1 : 122
Digit 2 : 221
Digit 3 : 322
Digit 4 : 414
Digit 5 : 513
Digit 6 : 632
Digit 7 : 713

Susunan data terakhir setelah mengalami proses pengelompokkan dari digit terakhir hingga digit pertama adalah seperti berikut :

122 221 322 414 513 632 713

Untuk proses secara descending maka kelompok digit dari ats ke bawah dibalik susunannya dari nilai yang terbesar menuju ke nilai yang terkecil.

8. Metode MergeSort

Metode MergeSort, sesuai dengan namanya mempunyai langkah-langkah proses memecah

susunan data menjadi sekecil mungkin kemudian masing-masing pasangan data yang saling berdekatan dibandingkan dan dipertukarkan tempatnya jika data yang disebelah kiri lebih besar. Setelah itu data digabungkan dan dibandingkan dengan pasangan data di sebelahnya yang juga dikenai proses yang sama. Begitu seterusnya proses tersebut diulang sehingga akhirnya diperoleh susunan data yang urut.

Contoh :

Data awal :
322 713 122 414 632 221 513

Susunan data di atas dibagi menjadi dua bagian yaitu seperti berikut ini :

322 713 122 414 632 221 513 …

Karena jumlah data adalah ganjil maka supaya jumlahnya menjadi genap ditambahkan data kosong di susunan data terakhir.
Kemudian masing-masing kelompok dipecah kembali menjadi dua bagian lagi sehingga hasilnya menjadi seperti berikut ini :

322 713 122 414 632 221 513 …

Selanjutnya setelah dibagi menjadi dua bagian lagi masing-masing data akan berdiri sendiri tanpa kelompok seperti berikut ini :

332 713 122 414 632 221 513 …

Masing-masing data dibandingkan dengan data pasangan di sebelahnya, ditukar tempat bila data sebelah kiri lebih besar dan kemudian hasilnya digabungkan, sehingga tampak susunan menjadi seperti berikut :

322 713 122 414 221 632 513 …

Hasil penggabungan kemudian diperbandingkan dengan pasangan hasil penggabungan di sebelahnya sehingga susunan data berubah menjadi seperti berikut :

122 322 414 713 221 513 632 …

Hasil pengurutan dan penggabungan masing-masing kelompok kemudian dibandingkan dan digabungkan setelah diurutkan, hasilnya menjadi seperti berikut :

122 221 322 414 513 632 713

Pengurutan secara descending cukup dengan mencari nilai yang lebih besar untuk ditukar tempatnya di sisi sebelah kiri saat dibandingkan.

9. Metode QuickSort

Sesuai dengan namanya, metode QuickSort ini merupakan suatu metode yang paling cepat dalam prosesnya. Metode ini menggunakan cara membagi suatu susunan data menjadi dua bagian, dengan nilai data awal sebagai nilai tengahnya. Sehingga akan diperoleh kelompok data di sebelah kiri yang lebih kecil dari nilai tengah dan kelompok data di sebelah kanan yang lebih besar dari nilai tengah. Selanjutnya masing-masing kelompok, baik kiri maupun kanan diperlakukan seperti langkah awal tersebut. Susunan data akan selalu terbagi menjadi dua bagian sampai akhirnya diperoleh susunan data yang urut. Yang perlu diperhatikan adalah bahwa sewaktu menempatkan kelompok data di sebelah kiri, atau yang lebih kecil dari nilai tengah, pembacaan data dimulai dari sisi kanan. Sebaliknya untuk data yang lebih besar dari nilai tengah pembacaan data dimulai dari sisi kiri. Mungkin untuk lebih jelas dapat diilustrasikan sebagai berikut :

Contoh :

Data awal :
30 70 10 40 60 20 50

Tentukan nilai data awal 30 sebagai nilai tengah, kemudian susun data di sebelah kiri nilai tengah yaitu data yang nilainya lebih kecil, dibaca mulai dari susunan data paling kanan. Sebaliknya untuk mengisi susunan data yang lebih besar dari nilai tengah dibaca mulai dari susunan data paling kiri. Hasil dari proses tersebut adalah sebagai berikut :

30 70 10 40 60 20 50

(10 20) **30** (70 40 60 50)

Berikutnya untuk masing-masing kelompok data sisi kiri dan sisi kanan dilakukan cara yang sama

seperti langkah awal. Nilai data 10 menjadi nilai tengah sisi kiri dan nilai data 70 menjadi nilai tengah sisi kanan, sehingga akan diperoleh hasil sebagai berikut :

10 20 **30** (**70** 40 60 50)
10 20 **30** (**40** 60 50) **70**

Hasil tersebut di atas diperoleh karena tidak ada nilai data yang lebih kecil dari 10 dan tidak ada pula nilai data yang lebih besar dari 70.
Di sisi kiri nilai data 20 menjadi nilai tengah sekaligus nilai akhir yang diurutkan dari sisi kiri karena hanya tinggal satu data tersebut. Untuk kelompok sisi kanan nilai data 40 menjadi nilai tengah berikutnya dengan hasil seperti berikut :

10 20 30 40 (**60** 50) **70**

Tinggal satu langkah lagi dimana nilai data 60 menjadi nilai tengah sekaligus merubah posisi tempat dengan nilai data 50, sehingga hasilnya menjadi seperti berikut :

10 20 30 40 50 **60 70**

Urutan data secara descending dapat diperoleh dengan membalik posisi dengan menempatkan data yang lebih besar di sebelah kiri nilai tengah dan data yang lebih kecil di sebelah kanan nilai tengah.

KESIMPULAN

Teknik-teknik pengurutan data memang cukup beragam, namun demikian tentunya dalam prosesnya ada metode yang tercepat seperti Quicksort dan metode yang dianggap lambat seperti misalnya Bubblesort. Tentunya masing-masing metode mempunyai kelebihan dan kelemahan sendiri sehingga hampir semua metode yang telah dibahas sebelumnya di atas masih dapat diterapkan tergantung dari permasalahan yang sedang dihadapi. Paling tidak dengan pemahaman algoritma secara sederhana tersebut lebih mudah dalam mencari metode yang tepat dalam menyelesaikan setiap permasalahan yang menyangkut proses pengurutan data.

DAFTAR PUSTAKA

1. Amsbury, W., (1995), "*Data Structures : From Arrays to Priority Queues*"
2. Carrano,F.M., Helman P., Veroff R., (1993), "*Data Structure and Problem Solving with Turbo Pascal*", Addison-Wesley Publishing
3. Flamig, Bryan, (1993), "*Practical Data Structures in C++*"
4. Hariyanto, B., (2000), "*Struktur Data*", Informatika, Bandung
5. Horowitz, E., Sahni, S., (1983), "*Fundamentals of Data Structures*"
6. Korsh, J.F., Garrett,L.J., (1994), "*Data Structures, Algorithms and Program Style Using C"*
7. Lipschulz, Seymor., (1992), "*Data Structures*", International Edition Schaum's Outline Series
8. Tanembaum, Aaron M., Moshe J.A., (1993), "*Data Structures Using Pascal*", Prentice-Hall Editions
9. Weiss, M.A., (1992), "*Data Structures and Algorithm Analysis in C* "
10. Wirth, Niklaus, (1997), "A*lgoritma + Struktur Data = Program"*, Andi Offset, Yogyakarta